

# Instalación y configuración de un Flujo de Diseño Digital

Leandro Marsó

Córdoba

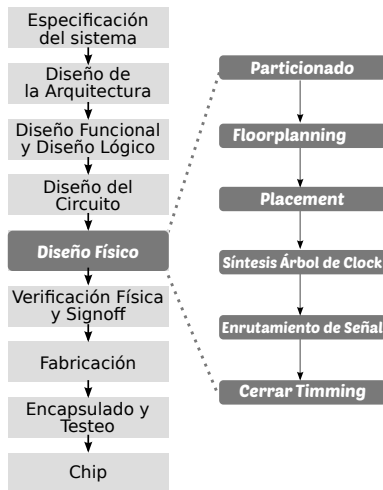
*elleandro@gmail.com*

18 de abril de 2015

# Contenido

- 1 Flujo físico
- 2 Instalación y prueba

# Diagrama de un flujo de diseño físico



# Instalación del flujo digital

Para instalar todas las herramientas, seguir las instrucciones detalladas en el repositorio que vamos a utilizar:

`https://github.com/elleandroculia/dflow-doc`

Asegurarse que todos los programas se hayan instalado correctamente.

# Descargar los ejemplos y la configuración

Descargamos todos los archivos de configuración para hacer funcionar todo el flujo con un diseño de prueba.

En un directorio cualquiera y desde la consola hacemos:

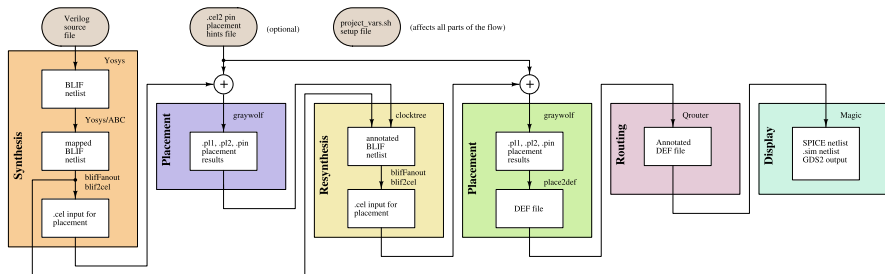
```
git clone https://github.com/elleandroculia/dflow-doc.git
```

Probar el ejemplo:

```
cd ejemplos/test  
qflow synthesize place buffer route display map9v3
```

El resultado final se mostrará en una ventana, donde podremos ver el circuito.

## Flujo de diseño digital



# Design kit

Por cada tecnología de fabricación, las herramientas necesitan un conjunto de archivos para cada etapa del diseño. Por ejemplo, archivo de configuración de DRC, un archivo con parámetros para hacer estimaciones de *timing* y potencia, librerías estándar, etc. Nuestra herramienta trae instalada 2 design kits: OSU035 y OSU05

## Síntesis de RTL a compuertas

**Qflow** utiliza la herramienta **yosys** para generar un netlist verilog con sólo celdas estándar instanciadas, que funcionalmente sea equivalente al descrito a nivel de RTL.

Qflow genera un script para síntesis completo en `source/nombreDelModulo.ys`. Para modificarlo, copiarlo y cambiarle el nombre y en el archivo `project_vars.sh` poner la opción **yosys\_options** a `s nuevoScript.ys`".



# Floorplan y placement con **graywolf**

## Herramienta de *placement*

En **Qflow** se utiliza **graywolf**. El resultado que produce esta herramienta es un *layout* en formato DEF, que representa al circuito realizado con celdas estándar instanciadas en el plano y los pines de entrada y salida.

# Archivos para guiar la herramienta

Para guiar el resultado de la herramienta, editar los archivos:

- *nombreDelMódulo.ce1*
- *nombreDelMódulo.ce12*
- *nombreDelMódulo.par*.

Por ejemplo, luego de la primera prueba podemos encontrar el archivo *map9v3.par* en el directorio *layout*:

## Modificar la cantidad de filas del floorplan

Si descomentamos la línea:

```
# GENR*numrows          : 6
```

nuestro *floorplan* tendrá 6 filas de celdas.

Descargar la documentación completa de Timberwolf:

<https://github.com/rubund/timberwolf/raw/master/doc/TimberWolf.pdf>

Aclaración: **Qflow** utiliza **graywolf** que es un fork de **Timberwolf 1.3**, del cual sólo usa las herramientas de *floorplan* y *placement*.

# DEF/LEF

**DEF:** Desing Exchange Format

**LEF:** Library Exchange Format

El formato es el mismo para ambos, pero cuando se utiliza para describir un bloque, se llama DEF, y para una librería de celdas estandars se denomina LEF. Se puede encontrar un archivo .lef con las reglas y definiciones de la tecnología, y otro .lef con la librería de celdas estándar.

Este formato es lo que usa el ruteador y el placer (entre otras herramientas) para conocer aspectos físicos de las celdas y los bloques. Tiene información del layout y de la tecnología.

# DEF/LEF

DEF: Desing Exchange Format

LEF: Library Exchange Format

¿Qué datos nos brinda este formato?

- Información sobre DRC útil para el ruteador, por ejemplo la separación mínima entre cables, máximo ancho de pista, capas de obstrucción, Antenna Rules.
- Básicamente es información de layout mas alguna información extra útil para el ruteador. Orientación de la celda o macrocelda.

# Modelos de timing y potencia

## NLDM: Non-Linear Delay Model

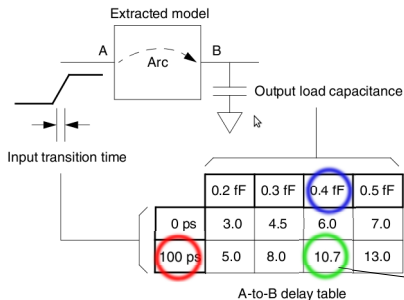
Es una forma de expresar los Delay, Timing Checks y Output Slew (tiempo de transición) de bloques. Asume carga capacitiva pura, utilizando capacidad equivalente en el caso de que la resistencia no es despreciable.

Para guardar esta información de las celdas, se utiliza el formato **Liberty**. Pero también se puede usar para caracterizar en tiempo y potencia bloques mas grandes, y poder usarlos como una caja negra.

# Archivo .lib (Liberty)

- El archivo .lib contiene información de timing y potencia de todas las celdas estándares, y también puede incluir la función de la celda, entre otras propiedades.
- Es un estandar abierto y suele venir incluido con la librería de celdas estándar.
- Es utilizado por la herramienta de síntesis **yosys** y la herramienta de STA (Static Timing Analysis) **vesta**

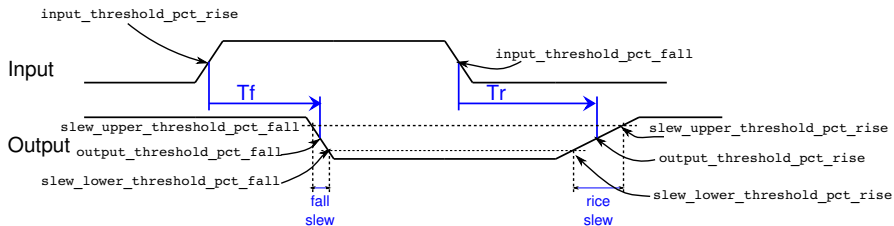
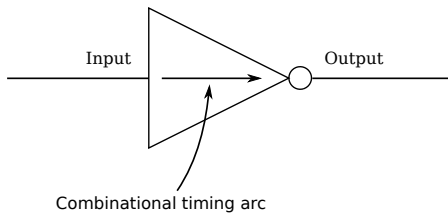
## Formato y significado del archivo .lib



```
# Primero definimos la tabla, se asigna
# cualquier valor a los indices
/* 2-D table template f(in_trans, out_cap) */
lu_table_template( 2x4-template ) {
    variable_1 : input_net_transition;
    variable_2 : total_output_net_capacitance;
    index_1 ( " 0.0000, 1.0000 " );
    index_2 ( " 0.0000, 2.0000, 3.0000, 4.0000 " );
}
```

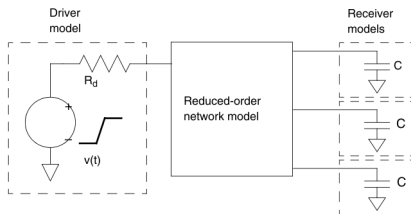
```
cell_rise( 2x4-template ){
    index_1 ( "0.000000, 0.100000";
    index_2 ( "0.002, 0.003, 0.004 0.005";
    values ( "0.300, 0.450, 0.600, 0.700", \
            "0.500, 0.800, 1.070, 1.300";
}
```





# Modelos de timing y potencia

## NLDM: Non-Linear Delay Model



Se especifica los tiempos según una tabla de 2 entradas:

- El tiempo de transición de la fuente de tensión a la entrada (driver)
- Capacidad de carga a la salida (receptor)

# Ruteador

**Qflow** utiliza el ruteador **qrouter**

En el archivo de configuración `project_vars.sh` encontramos dos variables que influyen en el comportamiento del ruteador:

**qrouter\_options**, **via\_stacks** y **via\_patterns**

Otras opción importante es la de especificar los metales permitidos para rutear.

# Fin